DETAILED ACTION

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee. Authorization for this examiner's amendment was given in a telephone interview with Benjamin Keim on 01/12/2010.

Claims 11-12, 18-19, 26, and 38 have been amended as follows:

Claim 11:

[[A]] One or more computer ~~readable~~ storage media ~~medium~~ having a tangible component including machine readable instructions for implementing the method as defined in Claim 1.

Claim 12:

In a managed code environment, a method implemented on a computing device having instructions stored on a computer-readable storage media and executable by a processor, comprising:

simulating calling from one assembly to another for which a permission set is required, wherein the simulation comprises one or more simulated stack walks that include two or more of the assemblies, each assembly being managed code in a library, wherein an execution of each assembly is statically simulated without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data, and wherein the simulated stack walk comprises:

entering a public entry point of a method in the assembly;

gathering a permission set for the method in the assembly;

determining whether the method in the assembly calls another method in the assembly or in an another assembly;

for each called method:

gathering a permission set for the another method called by the method in the assembly; and

determining whether the another method calls a subsequent method in the assembly or in the another assembly; and

creating a union of the gathered permission sets;

repeating the calling for each assembly in the managed code and for all possible execution paths of the managed code;

repeating the entering for each public entry point in the library;

finding the union of the permission sets corresponding to each call; and deriving security requirements for execution paths corresponding to the assemblies by using the union of the gathered permission sets across the execution paths corresponding to the one or more assemblies, wherein the union estimates the security requirements that will be triggered against the assemblies during an actual execution of the assemblies and whether a security exception will be triggered during the actual execution.

Claim 18:

[[A]] One or more computer ~~readable~~ storage media ~~medium~~ having a tangible component including machine readable instructions for implementing the method as defined in claim 12.

Claim 19:

One or more computer storage media having a tangible component comprising instructions that, when executed by a processor, perform a static simulation of an execution of every data and control flow for managed code from which an estimate is derived of the minimum security requirements needed to dynamically execute the managed code without triggering a security exception, the instructions comprising:

simulating, statically, one or more stack walks for each data and a control flow for the managed code, wherein the managed code corresponds to one or more assemblies, wherein the one or more stack walks comprise two or more of the assemblies, and

finding a set of required permissions for each execution path by the stack walks, wherein each call in each execution path has a corresponding permissions set, wherein each assembly has one or more execution paths representing a different data and control flow, and wherein the simulated stack walk comprises:

entering a public entry point of a method in an assembly;

gathering a permission set for the method;

determining whether the method calls another method;

for each called method:

gathering a permission set for the called method; and

determining whether the called method calls a subsequent method;

and

creating a union of the gathered permission sets; and

deriving the security requirements for execution paths corresponding to the two or more assemblies by using the union of the gathered permission sets, wherein the union estimates the security requirements that will be triggered against the two or more assemblies during an actual execution of the two or more assemblies.

Claim 26:

An apparatus comprising:

means for processing;

means for storing information in memory coupled to the means for processing;

virtual machine means, stored in the memory, in a managed code portion, for operating a plurality of assemblies in managed code, wherein the managed code is a managed shared library or an executable and is in the managed code portion;

execution engine means, in a native code portion, for executing the virtual machine means;

means, in the native code portion, for providing an operating system;

means for making a call in the managed code portion for access by one assembly to another assembly for which a permissions set is required;

means in the managed code portion for gathering the permissions set from each call;

means in the managed code portion for deriving a union of the gathered permissions sets;

means in the managed code portion for statically simulating the execution of all possible execution paths for the managed shared library or the executable without actually running a corresponding managed code, to derive therefrom the derived union of the gathered permissions sets wherein the means for simulating the execution performs, for each execution path, one or more simulated stack walks that each include a plurality of assemblies, and wherein the one or more simulated stack walks comprise:

means for entering a public entry point of a method in the assembly;

means for gathering a permission set for the method;

means for determining whether the method calls another method;

for each called method:

means for gathering a permission set for the called method;

means for determining whether the called method calls a subsequent method; [[and]]

means for repeating the previous gathering and determining until any gathered permission set is duplicative; and

means for creating a union of the gathered permission sets; and

means for deriving security requirements for execution paths corresponding to the plurality of assemblies by using the union of the gathered permission sets across the execution paths corresponding to the plurality of assemblies, wherein the union estimates whether a security exception will be triggered during an actual execution of the assemblies.

Claim 38:

A computing device comprising:

a processor;

a memory coupled to the processor;

a managed code portion stored in the memory;

a native code portion stored in the memory; and

an application program in the managed code portion comprising logic configured to:

statically simulate the execution of all possible calls from one assembly to another assembly for all possible execution paths of the managed code without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data, wherein each assembly call has a corresponding permissions set, wherein the simulation of the execution comprises one or more simulated stack walks that each include a plurality of [[the]] assemblies, and wherein the one or more simulated stack walks comprise:

a public entry point of a method in the assembly;

a permission set for the method;

a determination of whether the method calls another method;

for each called method:

a permission set for the called method;

a determination of whether the called method calls a subsequent method; and

a totality of permission sets such that any subsequent permission set is duplicative; and

a union of the permission sets;

derive a union of the permissions sets from each assembly call; and

derive security requirements for execution paths corresponding to the plurality of assemblies by using the union of the permission sets across the execution paths corresponding to the plurality of assemblies, wherein the union estimates the security requirements that will be triggered against the one or more assemblies during an actual execution of the assemblies.

Allowable Subject Matter

2. Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 are allowed.

3. The following is an examiner's statement of reasons for allowance:

4. As noted above, the Examiner agrees with the Applicant's arguments on pages 22-23 of the Remarks filed on 10/05/2009, specifically that the prior art does not teach "the execution of each assembly is statically simulated without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data" and "the simulated stack walk comprises: entering an execution path corresponding to a static simulation of execution of the assembly; entering a public entry point of a method in the assembly; gathering a permission set for the method in the assembly; determining whether the method in the assembly calls another method in the assembly or in an another assembly; gathering a permission set for the another method called by the method in the assembly; and creating a union of the gathered permission sets". The Examiner was unable to find a teaching, suggestion, or motivation that would render the above limitations obvious. Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 are, therefore, novel and not obvious.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to TRANG DOAN whose telephone number is (571)272-0740. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William R. Korzuch can be reached on (571) 272-7589. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Trang Doan/
Examiner, Art Unit 2431

/William R. Korzuch/
Supervisory Patent Examiner, Art Unit 2431